

2019-07-25 DSpace 7 Working Group Meeting

- [Date](#)
- [Agenda](#)
- [Attendees](#)
- [Current Work](#)
 - [Tickets / PRs In Progress](#)
 - [PRs Needing Review](#)
 - [PRs Merged this week!](#)
 - [Blocked](#)
 - [Delayed / Needs Discussion](#)
- [Notes](#)

Date

25 Jul 2019 from [14:00-15:00 UTC](#)

Location: <https://duraspace.zoom.us/my/dspace> (Meeting ID: 502 527 3040).

- More connection options available at [DSpace Meeting Room](#)

Agenda

- (15 mins) Developer Stand Up - Developers give brief updates on their effort (or their team's effort).
 - Update/see "Current Work" section below based on your status. *Please feel free to update prior to meeting.*
 - Please highlight any new work (needing reviews/testing), any blockers (for you), and any discussion topics you may have.
- (30 mins) General Discussion Topics
 1. Revisiting major undecided "gaps" in our [Development Planning Spreadsheet](#). These two features affect a larger number of outstanding tasks.
 - a. **Authorization display in UI:** How to pass Authorization rights (i.e. logged in user's access rights) from REST API to Angular? See for example: <https://github.com/DSpace/dspace-angular/issues/393>
 - i. Can this be achieved via passed HAL "_links" (e.g. the existence of an "edit" link in REST response means you must have Edit rights)?
 - b. **Scripts & Processes Endpoint:** <https://github.com/DSpace/Rest7Contract/pull/17>
 - i. How do we move this forward? In our [Jan 24](#) meeting, Tim noted a risk of "scope creep" with this feature suggestion. This idea has been tabled since then.
 2. Improved Estimation Strategy for Beta release. (Estimates should include time for discussion / code reviews, also should be done by multiple developers)
 - a. [Development Planning Spreadsheet](#) needs to be updated, so that we can determine what (outstanding) tasks need estimation.
 - b. Will be using [Wideband Delphi estimation technique](#)
 3. I18n status update
- (15 mins) Planning for next week
 - Assigning PRs for Review
 - Next tasks from [Development Planning Spreadsheet](#)

Attendees

- [Art Lowel \(Atmire\)](#)
- [Andrea Bollini \(4Science\)](#) - On holiday, will return on Aug 8 meeting.
- [Tim Donohue](#)
- [Heather Greer Klein](#)
- [Lieven Droogmans](#)
- [Mark H. Wood](#)
- [Giuseppe Digilio \(4Science\)](#)
- [Ben Bosman](#)
- [Pascal-Nicolas Becker](#)
- [Chris Wilper](#)
- [Terrence W Brady](#)
- [Pablo Prieto](#)
- [Paulo Graça](#)
- [Bill Tantzen](#)

Current Work

Legend for status icons

★ = Highest Priority tasks (please prioritize these reviews/tasks over others). Currently this lists tasks/features that need to be completed for Preview Release.

★ = Priority for OR2019 (Preview #2 Release).

blocked URL = review done (this week), changes were requested.

blocked URL = review done, approved.

⚠ = review done, merge conflict or other minor changes requests

Tickets / PRs In Progress

1. (Angular) Adding Accessibility via Travis CI <https://github.com/Dspace/dspace-angular/pull/356> (work in progress) (*Lower priority*)
2. blocked URL (Angular Bug) <https://github.com/Dspace/dspace-angular/issues/368> (Art Lowel (Atmire))
3. (REST Contract) Edit Homepage news: <https://github.com/Dspace/Rest7Contract/pull/45> (Ben Bosman - has outstanding questions/comments) (*Lower priority*)
4. (REST) DS-4043: Revisit the security layer of the submission (work in progress) Andrea Bollini (4Science)
5. blocked URL (REST Contract) Group and eperson management: <https://github.com/Dspace/Rest7Contract/pull/41> (Waiting on updates from Ben Bosman)
6. blocked URL (REST) Pagination issues on Items findAll - <https://github.com/Dspace/Dspace/pull/2406> (Waiting for confirmation of proposed approach by Andrea Bollini (4Science) from Kevin Van de Velde (Atmire) and any other interested/available)
7. (Angular) Transfer to .po message format - Initial PR: <https://github.com/Dspace/dspace-angular/pull/366> (Paulo Graça blocked URL, blocked URL Tim Donohue , Art Lowel (Atmire))

PRs Needing Review

1. (REST Contract) Collecting statistics - <https://github.com/Dspace/Rest7Contract/pull/63> (Dimitris Pierrakos, Andrea Bollini (4Science) - ✖ change s requested, Ben Bosman)
2. (REST Contract) OAI Harvesting configuration - <https://github.com/Dspace/Rest7Contract/pull/66> (Paulo Graça, Tim Donohue - ⚠ feedback provided)
3. (REST) Item Mapper functionality: <https://github.com/Dspace/Dspace/pull/2282> (Tim Donohue - ⚠ feedback provided, blocked URL Ben Bosman)
4. (REST) Pagination bug with withdrawn items: <https://github.com/Dspace/Dspace/pull/2406> (Dimitris Pierrakos , Ben Bosman - Feedback provided)
5. (Angular) (Entities) Deleting relationships: <https://github.com/Dspace/dspace-angular/pull/402> (Paulo Graça - will test again, Tim Donohue)
6. (Angular) Move Item Component: <https://github.com/Dspace/dspace-angular/pull/335> (Giuseppe Digilio (4Science) reviewed and provided feedback, Tim Donohue)
7. (Angular) Item-Collection Mapper: <https://github.com/Dspace/dspace-angular/pull/348> (**NEEDS REVIEWERS**)
 - a. Dependent on PR#2282 (above)
8. (Angular) Shibboleth integration support (*WORK IN PROGRESS*): <https://github.com/Dspace/dspace-angular/pull/429> (**Julius has updated the PR. Ready for RE-REVIEW**) (Giuseppe Digilio (4Science), Fernando FCT/FCCN, Paulo Graça)
 - a. 4Science will enable Shibboleth on public demo.
9. (Angular) Submission Miscellaneous fixes: <https://github.com/Dspace/dspace-angular/pull/432> (Art Lowel (Atmire) blocked URL feedback provided, Julius Gruber)
10. (Angular) Replace model factories <https://github.com/Dspace/dspace-angular/pull/434> (✔ Tim Donohue - approved but with some feedback, Giuseppe Digilio (4Science))
11. (**MERGE - minor merge conflicts**) (Backend) Upgrading to Handle Server v9: <https://github.com/Dspace/Dspace/pull/2394> (✔ Mark H. Wood, blocked URL Ben Bosman)
12. (**MERGE AFTER MEETING**) (Backend) One Webapp Phase 2: Rename "dspace-spring-rest" to "dspace-server-webapp": <https://github.com/Dspace/Dspace/pull/2459> (✔ Mark H. Wood, ✔ Alexander Sulfrian, ✔ Terrence W Brady)

PRs Merged this week!

1. ✔ (REST) [DS-4266] bitstream format registry <https://github.com/Dspace/Dspace/pull/2442>
2. ✔ (Angular) Add missing ResourcePolicyService: <https://github.com/Dspace/dspace-angular/pull/427>
3. ✔ (Backend) Update email templates to use Velocity (for richer, powerful templates): <https://github.com/Dspace/Dspace/pull/1992>

Blocked

1. (Blocked PRs go here)

Delayed / Needs Discussion

1. Initial Performance Testing from Chris.
 - <https://cwilper.github.io/dspace-perftest/>
2. (REST Contract) Edit Homepage News: <https://github.com/Dspace/Rest7Contract/pull/45>

- a. Delayed until after Preview release. General agreement ([in meeting on March 21, 2019](#)) that storing HTML in metadata fields is not really ideal behavior. Metadata (from a librarian standpoint) tends to be free of format-related markup (as that allows for easier sharing, understanding of metadata. Currently Community & Collection homepage information is HTML-based and *is stored in metadata* that is appropriate for a minor subset of information (like the title) but it is better to move large/rich text to bitstreams.
 - b. Proposal here is to consider storing HTML-based markup (for Site, Community & Collection homepages) in Bitstream(s) associated with the object in question. May allow for more CMS-lite behavior in the future
 - c. Timeline for this is uncertain. Possibly in 7 or 8. May depend on how/whether it can be scoped.
3. (REST) Scripts & Processes endpoint: <https://github.com/DSpace/Rest7Contract/pull/17>
 4. Concurrency in DSpace 7 (or 8). What do we want to do when multiple editors are editing the same object? Needs further analysis regarding implementation details
 - a. We've decided ([in meeting on March 7, 2019](#)) to use ETags to implement concurrency. REST Contract notes on ETags: <https://github.com/DSpace/Rest7Contract#etags--conditional-headers>
 - b. ETags only update if the two fields match. If someone edits first, your edit would fail and you would get a 422 response (422?)
 - c. ETags seems to have broader support in other REST APIs. Recommended also by both Art and Andrea.
 5. Not discussed much, but could there be opportunities to use Travis CI + Docker Compose for testing of Angular??
 - Docs: <https://docs.travis-ci.com/user/docker/#using-docker-compose>
 - Blogpost on how to do it: <http://elliott.land/post/using-docker-compose-on-travis-ci>
 - An example travis.yml file: <https://github.com/Ortus-Solutions/docker-buildfiles/blob/master/.travis.yml>

Notes

- **Authorization display in UI:** How to pass Authorization rights (i.e. logged in user's access rights) from REST API to Angular? See for example: <https://github.com/DSpace/dspace-angular/issues/393>
 - After much discussion, we've realized there may not be a **single** solution to this problem. Though more analysis is necessary to determine that.
 - Tim has started a [Managing Authorization info in Angular UI](#) wiki page for early ideas/brainstorms. The initial ideas brought up in this discussion are noted there already
- **Scripts & Processes Endpoint:** <https://github.com/DSpace/Rest7Contract/pull/17>
 - Tim still worried about scope creep here. Asked for more info on the proposed backend solution
 - Main questions are around the proposed `/api/system/processes` endpoint in the REST Contract, as it seems to require several features that don't currently exist on the DSpace backend (Java API or database or Solr). Some of those seemingly non-existent features include:
 - A way of tracking currently running DSpace processes, their start-time, current status, runtime parameters, etc.
 - A way to assign UUIDs to all processes, and link those processes to DSpace user accounts (UUIDs)
 - A way to see the output of finished processes individually (currently most processes just log to the DSpace log file, not to individual log files)
 - Atmire will take this feedback back and try and clarify the proposal for how a backend implementation might work.
- I18n status update on <https://github.com/DSpace/dspace-angular/pull/366>
 - Art noted that after further analysis, it looks like the "ngx-translate" PO file loader (<https://github.com/biesbjerg/ngx-translate-po-http-loader>) seems unmaintained, and doesn't seem to have many users. It also seems to not be fully compatible with later versions of Angular, and doesn't work with Universal (SEO) yet.
 - This makes our ability to support PO files more complex...we'd have to fix this loader to work properly & with Universal.
 - The Angular project itself has taken a different direction altogether. It uses XLIFF (instead of PO). However, instead of managing translations on the client side, they are doing so on the server-side. So, if you want an application to support 5 different languages, you'd build/deploy the application 5 times (one for each language). If users want to switch language, they are actually switching apps (though their session info can be passed between apps).
 - Angular project claims this server side setup is much more performant for larger applications. Unclear if we'll see this sort of performance problem though.
 - Neither of these solutions sounds ideal at this point. We'd like to support common translation formats (e.g. PO or XLIFF) to make translations easier on translators. But, there's not a clear "winner" between these options.
 - For the time being, we will stick with JSON (default format of "ngx-translate"), but flatten our JSON keys (to make things easier for translators)
 - We will track the PO loader for ngx-translate. If it gets more movement/support, we'll move to using it.
 - We will also track performance of ngx-translate. If we find that it's not performing well, we may eventually need to move to the Angular project approach (server side)